

"Everybody should learn to program a computer, because it teaches you how to think."

# **TUJUAN PEMBELAJARAN**

Mampu menjelaskan dan mengimplementasikan Operator dalam pemrograman menggunakan IDE.

### POKOK MATERI

- 1. Operator Aritmatika
- 2. Operator Penugasan
- 3. Operator Pembanding
- 4. Operator Logika
- 5. Operator Bitwise
- 6. Operator Kondisional

### **URAIAN MATERI**

### A. Operator

Operator adalah sebuah simbol. Simbol yang digunakan untuk melakukan operasi tertentu. Misalnya: Kita ingin menjumlahkan nilai dari variabel x dan y, maka kita bisa menggunakan operator penjumlahan (+).

### x + y

Berdasarkan jumlah operand-nya, operator dibagi menjadi 3 jenis:

Updated. 20 June 2022

- 1) Operator Unary, mempunyai 1 operand
- 2) Operator Binary, mempunyai 2 operand
- 3) Operator Ternary, mempunyai 3 operand

Operand adalah nilai asal yang dipakai dalam sebuah proses operasi. Sedangkan Operator adalah instruksi yang diberikan untuk mendapatkan hasil dari proses tersebut.

	Operator	Type
Unary operator	+ +,	Unary operator
	+, -, *, /, %	Arithmetic operator
Binary operator	<, <=, >, >=, ==, !=	Relational operator
	&&,   ,!	Logical operator
	&,  , <<, >>, ~, ^	Bitwise operator
	=, +=, -=, *=, /=, %=	Assignment operator
Ternary operator	?:	Ternary or conditional operator

# B. Operator Aritmatika

Operator aritmatika adalah operator yang digunakan untuk operasi matematika.

Nama Operator	Simbol
Penjumlahan	+
Pengurangan	
Perkalian	*
Pembagian	1
Sisa Bagi	%

```
#include <stdio.h>

void main(){

   int a, b, c;

   printf("Inputkan nilai a: ");
   scanf("%i", &a);

   printf("Inputkan nilai b: ");
   scanf("%i", &b);

   c = a + b;

   printf("Hasil a + b: %i", c);
}
```

```
#include <stdio.h>

void main(){
    int a, b;
    printf("Inputkan nilai a: ");
    scanf("%i", &a);

    printf("Inputkan nilai b: ");
    scanf("%i", &b);

    printf("Hasil a + b: %i\n", a + b);
    printf("Hasil a - b: %i\n", a - b);
    printf("Hasil a * b: %i\n", a * b);
    printf("Hasil a / b: %i\n", a / b);
    printf("Hasil a / b: %i\n", a / b);
    printf("Hasil a % b: %i\n", a % b);
}
```

Pada operasi pembagian, 7/2 hasilnya adalah 3. Mengapa demikian?

### Bukannya seharusnya 3.5

Ini karena kita melakukan operasi terhadap tipe data integer. Bila kita ubah programnya menggunakan tipe data float, seperti berikut:

```
#include <stdio.h>

void main(){

   float a, b;

   printf("Inputkan nilai a: ");
   scanf("%i", &a);

   printf("Inputkan nilai b: ");
   scanf("%i", &b);

   printf("Hasil a / b: %f\n", a / b);
}
```

# C. Operator Penugasan

Operator penugasan (Assignment Operator) merupakan operator untuk memberikan tugas pada variabel. Biasanya untuk mengisi nilai.

Operator Penugasan terdiri dari:

Nama Operator	Simbol
Pengisian Nilai	=
Pengisian dan Penambahan	+=
Pengisian dan Pengurangan	-=
Pengisian dan Perkalian	*=
Pengisian dan Pembagian	/=
Pengisian dan Sisa bagi	%=
Pengisian dan shift left	<<=
Pengisian dan shift right	>>=
Pengisian dan bitwise AND	&=

# Nama Operator Simbol Pengisian dan bitwise OR Pengisian dan bitwise XOR ^=

```
#include <stdio.h>
void main(){
    int a, b;
    // pengisian nilai dengan operator =
    a = 5;
    b = 10;
    // pengisian sekaligus penambahan
    b += a; // ini sama seperti b = b + a
    printf("Hasil b += a adalah %d\n", b);
    // pengisian sekaligus pengurangan
    b -= a; // ini sama seperti b = b - a
    printf("Hasil b -= a adalah %d\n", b);
    // pengisian sekaligus perkalian
    b *= a; // ini sama seperti b = b * a
    printf("Hasil b *= a adalah %d\n", b);
    // pengisian sekaligus pembagian
    b /= a; // ini sama seperti b = b / a
    printf("Hasil b /= a adalah %d\n", b);
    // pengisian sekaligus sisa bagi
    b %= a; // ini sama seperti b = b % a
    printf("Hasil b %= a adalah %d\n", b);
```

Pada program tersebut, variabel b kita isi ulang dengan operator penugasan.

Sebagai contoh, operasi:

```
b += a
```

Sama seperti operasi:

b = b + a

Artinya kita akan mengisi nilai untuk b dengan nilai b ditambah nilai a.

# D. Operator Pembanding

Operator pembanding adalah operator untuk memabndingkan dua buah nilai.

Operator ini juga dikenal dengan operator relasi.

Operator pembanding terdiri dari:

Nama Operator	Simbol
Lebih Besar	>
Lebih Kecil	<
Sama Dengan	==
Tidak Sama dengan	!=
Lebih Besar Sama dengan	>=
Lebih Kecil Sama dengan	<=

Nilai yang dihasilkan dari operasi pembanding akan berupa true dan false. Pada bahasa C, nilai true akan samadengan 1 dan false akan samadengan 0.

```
#include <stdio.h>

void main(){
    int a = 4;
    int b = 3;

    printf("a = %d\n", a);
    printf("b = %d\n", b);

// operator pembanding
    printf("a > b = %d\n", a > b);
    printf("a < b = %d\n", a < b);
    printf("a >= b = %d\n", a <= b);
    printf("a <= b = %d\n", a <= b);
    printf("a == b = %d\n", a == b);
    printf("a != b = %d\n", a != b);
}</pre>
```

# E. Operator Logika

Kalau kamu pernah belajar logika matematika, pasti tidak akan asing dengan operator ini.

Nama Operator	Simbol di Java
Logika AND	&&
Logika OR	11
Negasi/kebalikan	<u>!</u>

Operator Logika digunakan untuk membuat operasi logika.

Misalnya seperti ini:

Pernyataan 1: A seorang programmer

Pernyattan 2: A menggunakan Linux

Jika ditanya, apakah A programmer yang menggunakan Linux?

Tentu kita akan cek dulu kebenarannya

Pernyataan 1: A seorang programmer = true.

Pernyattan 2: A menggunakan Linux = true.

Apa A programmer dan menggunakan Linux?

Pernyatan 1	Pernytaan 2	Logika AND
true	true	true
true	false	false
false	true	false
false	false	false

# Contoh Program:

```
#include <stdio.h>

void main(){
    int a = 1; // true
    int b = 0; // false

    printf("a = %d\n", a);
    printf("b = %d\n", b);

    // logika AND
    printf("a && b = %d\n", a && b);

    // logika OR
    printf("a || b = %d\n", a || b);

    // logika NOT
    printf("!a = %d\n", !a);
}
```

# F. Operator Bitwise

Operator bitwise merupkan operator yang digunakan untuk operasi berdasarkan bit (biner) dari sebuah nilai.

# Algorithm & Programming

Operator bitwise terdiri dari:

Nama Operator	Simbol
AND	&
OR	I
XOR	^
NOT/komplemen	~
Left Shift	<<
Right Shift	>>

Buat kamu yang sudah pernah belajar sistem bilangan dan sistem digital akan mudah memahami cara kerja operator ini.

Tapi buat kamu yang belum, perhatikan penjelasan berikut:

Misalkan kita punya nilai 6 dan 3. Nilai 6 dan 3 kemudian kita ubah ke dalam bentuk bilangan biner. Sehingga akan menjadi seperti ini:

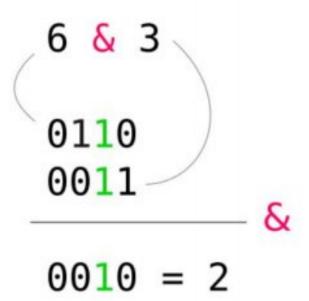
```
6 = 0110
3 = 0011
```

operator bitwise akan melakukan operasi berdasarkan biner-biner tersebut.

# 1. Bitwise AND (&)

Bitwise AND merupakan operasi bit berdasarkan logika AND, perhatikan gambar ini.

# **Bitwise AND**



Perhatikan bilangan biner untuk nilai 6 dan 3. Apabila kita melakukan operasi AND, maka akan menghasilkan bilangan biner baru. Kemudian biner yang dihasilkan tersebut diubah kembali ke dalam bilangan desimal. Maka hasilnya adalah 2.

```
#include <stdio.h>

void main(){
   int a = 6;
   int b = 3;

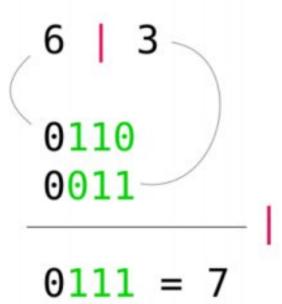
   printf("a & b = %d\n", a & b);
}
```

### 2. Bitwise OR ( | )

Operator bitwise OR juga sama seperti bitwise AND.

Operator bitwise OR akan menghasilkan nilai false atau 0 saat keduanya false.

# **Bitwise OR**



```
#include <stdio.h>

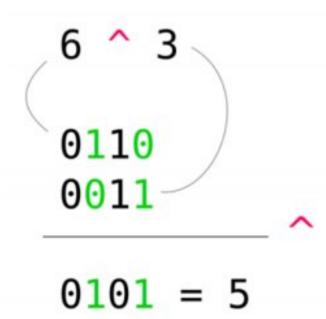
void main(){
   int a = 6;
   int b = 3;

   printf("a | b = %d\n", a | b);
}
```

### 3. Bitwise XOR (^)

Operator XOR (Exclusive OR) akan menghasilkan nilai 1 saat kedua nilai tidak sama.

# **Bitwise XOR**



```
#include <stdio.h>

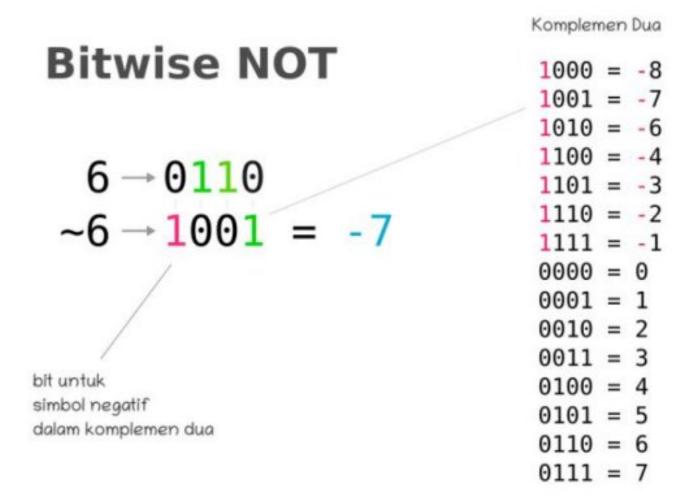
void main(){
   int a = 6;
   int b = 3;

   printf("a & b = %d\n", a ^ b);
}
```

# 4. Bitwise NOT (~)

Bitwise NOT dikenal juga dengan komplemen.

Operator ini akan menghasilkan nilai biner yang terbalik dari biner aslinya. Kemudian direpresentasikan dengan komplemen dua.



```
#include <stdio.h>

void main(){
   int a = 6;

   printf("~a = %d\n", ~a);
}
```

### 5. Bitwise Left Shift (<<)

Operator bitwise left shift akan menghasilkan nilai biner yang digeser ke kiri

```
6 << 1
```

Maka niai biner dari 6 akan digeser sebanyak 1 bit ke arah kiri.

```
Left Shift
6 = 0 \ 1 \ 1 \ 0
6 << 1 = 1 \ 1 \ 0 \ 0 = 12
geser 1 bit ke kiri
```

```
#include <stdio.h>

void main(){
   int a = 6;

   printf("a << 1 = %d\n", a << 1);
}</pre>
```

# 6. Bitwise Right Shift (>>)

Bitwise right shift sama seperti left shift. Perbedaanya terletak pada arahnya. Right shift akan menggeser bit ke arah kanan.

Contoh:

```
6 >> 1
```

Maka nilai biner dari 6 akan digeser ke arah kanan sebanyak 1 bit.

# Right Shift

$$6 = 0 \ 1 \ 1 \ 0$$
 $6 >> 1 = 0 \ 0 \ 1 \ 1 = 3$ 
geser 1 bit ke kanan

```
#include <stdio.h>

void main(){
   int a = 6;

   printf("a >> 1 = %d\n", a >> 1);
}
```

# 7. Operator Lain-lain

Selain dari operator yang kita bahas di atas, terdapat beberapa operator lain yang harus diketahui:

Nama Operator	Simbol	Keterangan
Alamat memori	&	untuk mengambil alamat memori
Pointer	*	untuk membuat pointer
Ternary	?:	untuk membuat kondisi
Increment	++	untuk menambah 1
Decrement		untuk mengurangi 1

Operator & jika digunakan pada satu variabel, akan berfungsi untuk mengambil alamat memori dari variabel tersebut. Dan operator \* kan berfungsi untuk membut sebuah pointer.

```
#include <stdio.h>

void main(){
    // membuat pointer
    int *a = 5;

    // ambil alamat memori dari pointer a
    printf("alama memori variabel a adalah %x\n", &a);
}
```

Berikutnya ada operator ternary untuk membuat kondisi. Cara kerjanya seperti percabangan if/else.



```
#include <stdio.h>

void main(){
   int a = 4;

   printf("a > 1 adalah %s", a > 1 ? "benar": "salah");
}
```

Terakhir ada operator increment dan decrement untuk menambah dan mengurangi nilai dengan 1.

```
#include <stdio.h>

void main(){
    int a = 4;

    // increment a
    a++;

    printf("a++ = %d\n", a);

    // increment a
    ++a;

    printf("++a = %d\n", a);

    // decrement a
    a--;

    printf("a-- = %d\n", a);

    // decrement a
    --a;

    printf("--a = %d\n", a);
}
```

Hasil outputnya:

```
a++ = 5
++a = 6
a-- = 5
--a = 4
```

Operator increment dan decrement dapat ditaruh di depan maupun di belakang variabel atau nilai. Operator ini biasanya digunakan pada blok perulangan.

### **LATIHAN**

 Buatlah program konversi detik kedalam bentuk jam, menit, beserta sisa detik.

Masukan jumlah waktu dalam detik: 3700

Hasil Output: 1 Jam, 1 Menit, 40 Detik

- Buatlah program kasir barang yang meminta inputan nama pembeli, nama barang, harga barang satuan, jumlah barang yang dibeli, dan harga total (harga barang \* jumlah barang).
- Nilai akhir dari praktikum pemrograman C ditentukan oleh empat nilai berikut:
  - a) Nilai Presensi bobot 10%
  - b) Nilai Praktek bobot 20%
  - c) Nilai UTS bobot 30%
  - d) Nilai UAS bobot 40%

Buatlah program untuk menghitung nilai akhir, jika diketahui Nilai Presensi= 85, Nilai Praktek=65, Nilai UTS=80 dan Nilai UAS= 75!

4. Buatlah sebuah program rental film yang menyewakan kesempatan menonton film dengan tarif 1 jam pertama sebesar Rp 15.000 Untuk tarif jam berikutnya adalah 50% dari tarif tetap 1 jam pertama. Hitung berapa total yang harus dibayar jika seseorang menonton film dengan durasi 3 jam?

# Algorithm & Programming

### REFERENCES

- 1. The C Programming Language. 2nd Edition
- 2. https://en.wikipedia.org/wiki/C\_(programming\_language)
- 3. https://en.wikipedia.org/wiki/Imperative\_programming
- 4. https://www.petanikode.com/tutorial/c/
- 5. https://www.learn-c.org/
- 6. https://www.tutorialspoint.com/cprogramming/index.htm
- 7. https://www.programiz.com/
- 8. https://www.dicoding.com/
- 9. https://data-flair.training/blogs/c-tutorials-home/